CLAIMS

1. A central processing unit comprising,

a first processor for executing program instructions

a program counter and registers shared by said first processor, said program counter containing an indicator of the address of the next instruction to be performed,

a function lookup unit sharing said program counter and registers, said function lookup unit having,

a lookup cache with a tag field, and a function indicator field, said lookup cache communicating with said program counter to determine the presence of a match between the contents of said tag field and said program counter indicator, upon finding a match between said tag field and said program counter indicator said function lookup unit assumes control of execution at said instruction indicated by said program counter and upon finding no match said first processor assumes control of execution at said instruction indicated by said program counter,

a reconfigurable combinational array having logic functions available which provide a mapping from a beginning state on entry to a block of code to an ending state on exit from said block of code, said functions having identifiers in said function indicator field, said reconfigurable combinational array receiving said function indicator identifiers from said lookup cache upon finding a match between said tag field and said program counter indicator, said functional indicator identifiers causing said function to be executed in place of said block of instructions starting at said program counter indicator.

2. The central processing unit of claim 1 also including an exception routine that provides logic synthesis for said block of instructions identified by said program counter in response to an attention signal from said function lookup unit, said exception function providing said synthesized function to said reconfigurable combinational array and a corresponding function indicator to said function indicator field.

3. The central processing unit of claim 1 in which said logic functions are preloaded into said reconfigurable combinational array before executing said program instructions.

4. The central processing unit of claim 2 in which said logic functions synthesized by said exception routine are created and loaded into said reconfigurable combinational array on the fly as program instructions are being executed.

5. The central processing unit of claim 2 having a counter for determining the number of times said block of instructions is executed, said exception routine being initiated when said counter reaches a threshold number of repetitions of said block of instruction executions.

6. The central processing unit of claim 2 in which said counter is a count field in said lookup cache.

7. The central processing unit of claim 1 in which said first processor provides an indication to said function lookup unit when said first processer overwrites the contents of a program counter address, said function lookup unit responds to said indication and searches for a matching program counter address in said lookup cache, upon finding a matching cache program counter address the cache makes a note that the cache lookup program counter address has been overwritten.

8. The central processing unit of claim 7 in which said noted presence of said matching cache program address causes the matched program address in said lookup cache to be inactivated.

9. The program counter of claim 1 in which said indicator of the address is an address.